Fast Liveness Using DJ Graphs

Dibyendu Das, IBM India
Ramakrishna Upadrasta, INRIA Saclay

Agenda

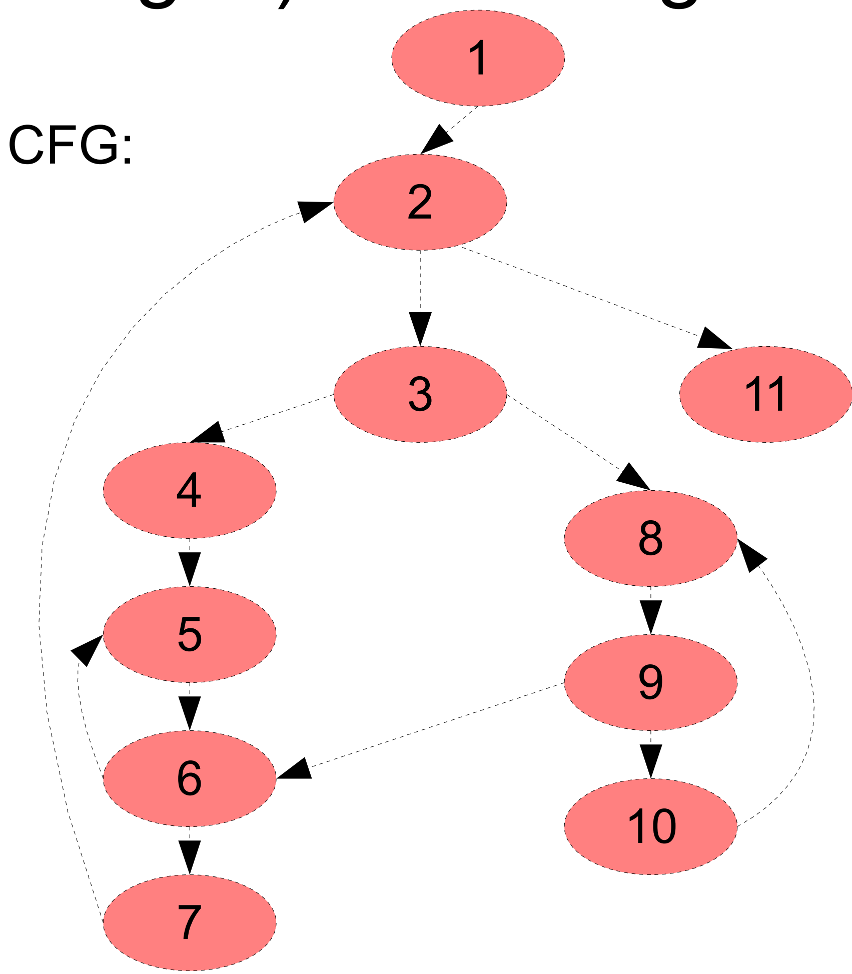
- DJ Graphs and Merge Sets
- Top-down Merge Set Computation
- Liveness Analysis – Boissinot et al
- Liveness Analysis using Merge Sets
- Conclusion

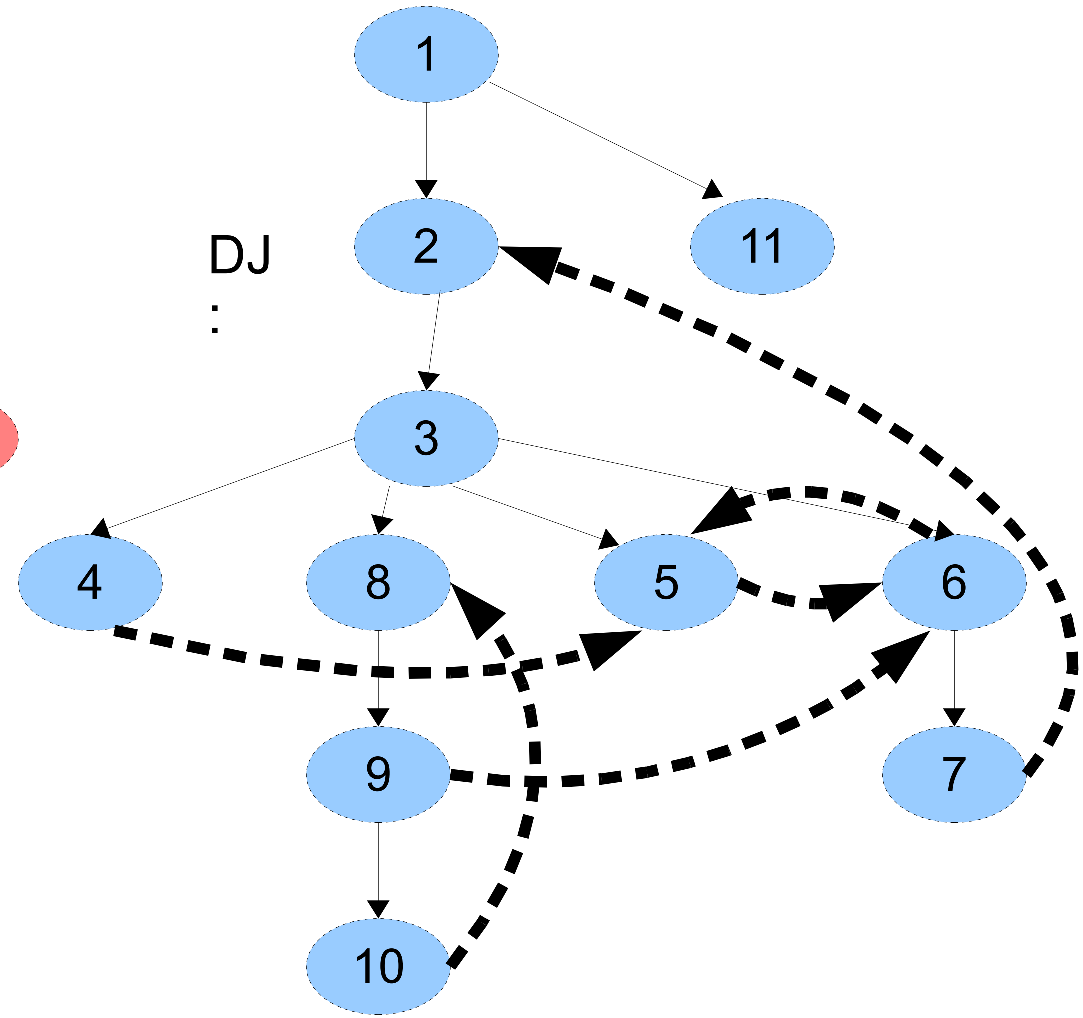
DJ Graphs

- For a CFG G=(V,E), an edge *e=(s,t)* is a J-edge when *s* does not strictly dominate *t*.
- J-edges are a subset of E
- A DJ Graph[Sreedhar POPL '95] $G_{DJ}=(V,E_{DJ})$ such that
  - $E_{DJ} = E_{D\text{-}tree} \cup J$
  - $E_{DJ} = E \cup E_{D\text{-}tree}$

DJ Graphs (contd ...)

- DJ Graphs consist of dominator edges (D-edges) and J-edges



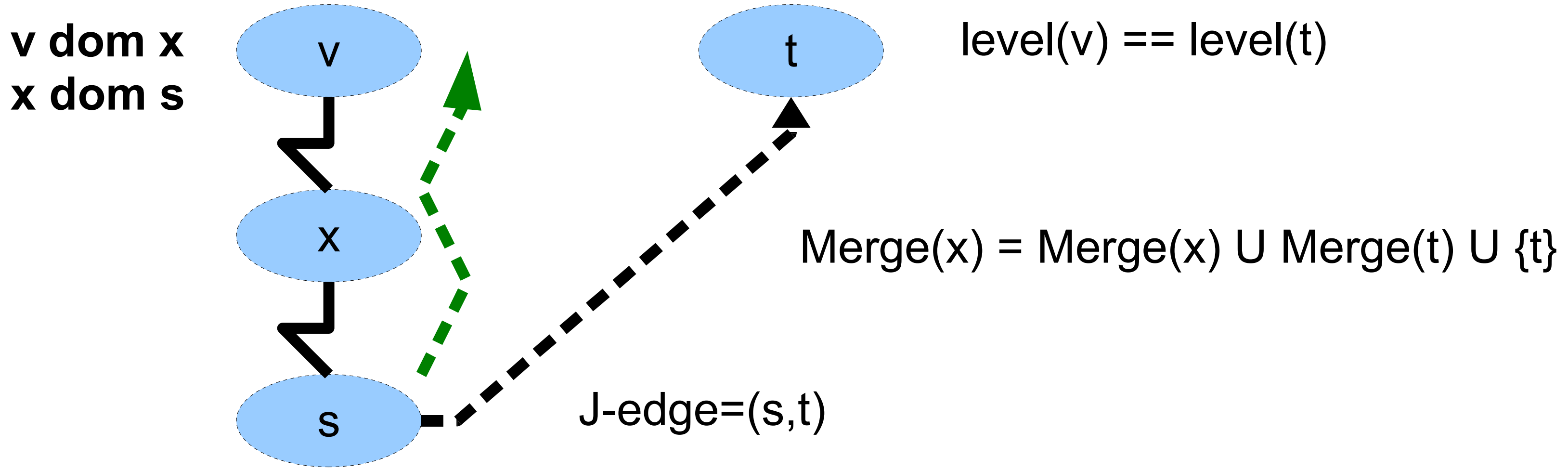


Merge Set

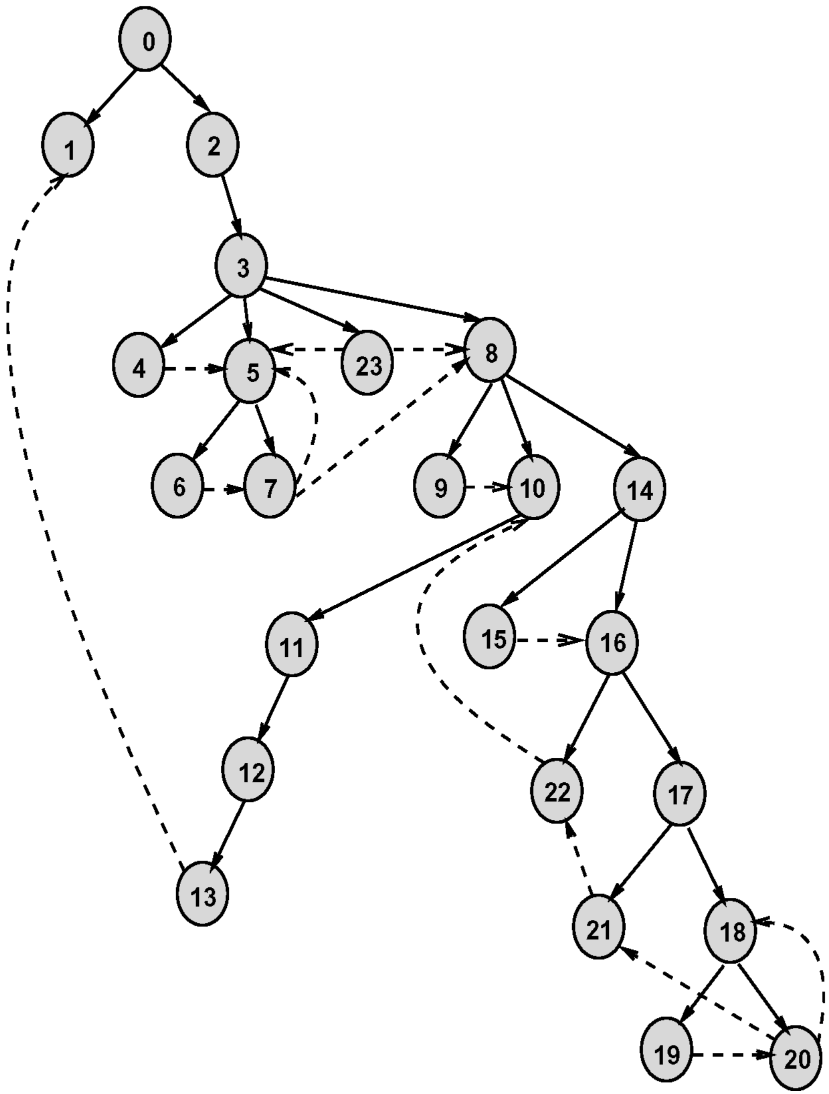
- **Definition**: Merge Set [Pingali et al JACM 2003] of a node n is the set of nodes, denoted as Merge(n), where a Φ needs to be placed if a variable is defined in n.
- Merge Sets of the nodes in a CFG can be computed, irrespective of where the definitions finally appear.
- The advantages are in re-using Merge(n) information when variables are defined in same/ similar nodes. As long as the CFG remains unchanged Φ-placement can re-use Merge(n) information.

Top-Down Merge Set Computation(TDMSC)

- The algorithm proceeds top-down on the DJ Graph starting at the start node and computing level-by-level [Das and Ramakrishna TOPLAS 2005]
- At each level we look for J-edge=(s,t) targets and update the merge set of each node **x** lying between the levels of the source and target in the dominator tree using ***Merge(x) = Merge(x)*** U ***Merge(t)*** U {***t***}. Implies ***Merge***(***x***)⊇***Merge***(***t***).



TDMSC -an example



Merge(4)	={5} U Merge(5);	{5,8}
Merge(5)	={5,8} U Merge(5) U Merge(8);	{5,8}
Merge(6)	={7} U Merge(7);	{5,7,8}
Merge(7)	={5,8} U Merge(5) U Merge(8);	{5,8}
Merge(9)	={10} U Merge(10);	{10}
Merge(14)	={10} U Merge(10);	{10}
Merge(15)	={16} U Merge(16);	{10,16}
Merge(16)	={10} U Merge(10);	{10}
Merge(17)	={22} U Merge(22);	{10,22}
Merge(18)	={18,21} U Merge(18) U Merge(21);	{10,18,21,22}
Merge(19)	={20} U Merge(20);	{10,18,20,21,22}
Merge(20)	={18,21} U Merge(18) U Merge(21);	{10,18,21,22}
Merge(21)	={22} U Merge(22);	{10,22}
Merge(22)	={10} U Merge(10);	{10}
Merge(23)	={5,8} U Merge(5) U Merge(8);	{5,8}

Advantages of TDMSC

- Can be applied to any arbitrary CFG – reducible or irreducible
- May require multiple passes(iterative) for the Merge sets to reach respective fixed points
- Analyses using SPEC CPU2000 shows that almost for 80% of cases a single top down pass suffices.

Liveness Analysis using Merge Sets

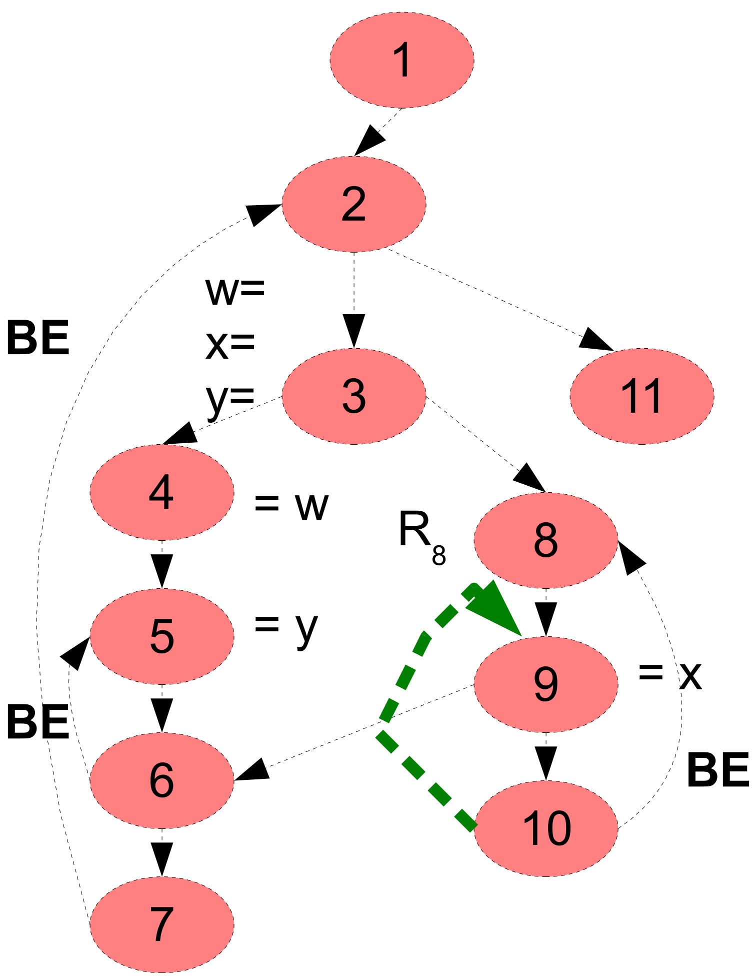
- Merge sets can also be used to compute fast liveness (proposed by Boissinot et al [CGO 2008])

Fast Liveness Analysis

- IsLiveIn(v,n): A variable **v** is live-in at node **n** if there exists a path from **n** to a **use** of **v** that does not pass through a **def** of **v**
- IsLiveIn uses the dominator tree and the $\mathbf{T_q}$ and $\mathbf{R_q}$ sets
- $\mathbf{T_q}$ – is the set of target nodes of back-edges in a CFG that "affect" node q
- $\mathbf{R_q}$ – is the set of nodes reachable from q in the back-edge-free CFG

Fast Liveness Using T_q and R_q – a motivating example

- IsLiveIn(10,x) ?

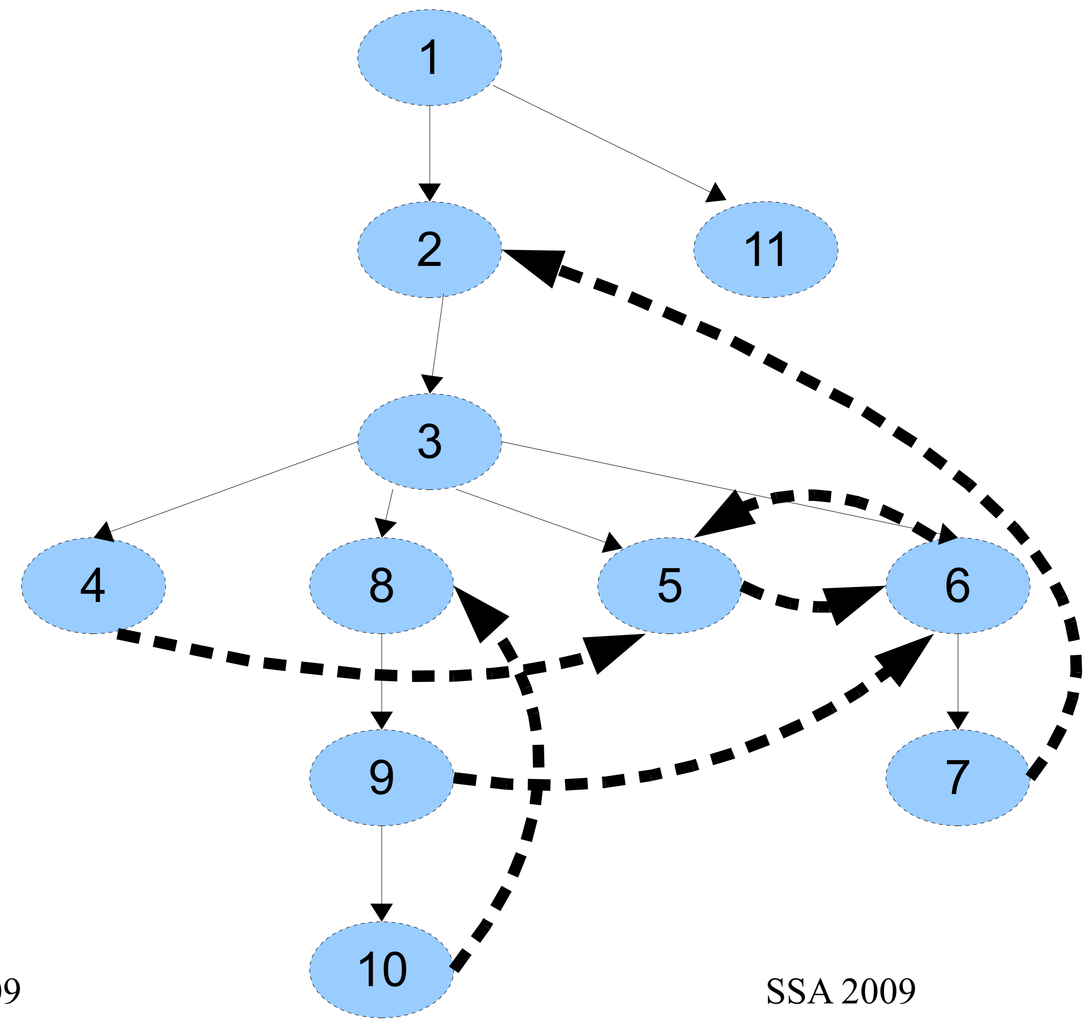


$T_1 = \{1\}$
$T_2 = \{2\}$
$T_3 = \{2,3\}$
$T_4 = \{2,4\}$
$T_5 = \{2,5\}$
$T_6 = \{2,5,6\}$
$T_7 = \{2,7\}$
$T_8 = \{2,5,8\}$
$T_9 = \{2,5,8,9\}$
$T_{10} = \{2,5,8,10\}$
$T_{11} = \{11\}$

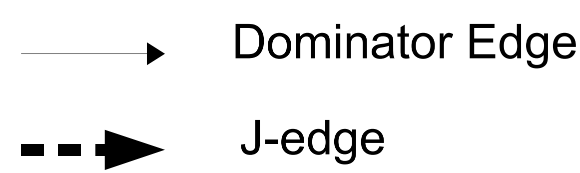
$R_9 = \{6,7,9,10\}$
$R_7 = \{7\}$

The DJ Graph and Merge Sets for the same example

Here are the merge sets for the same example:



Merge(1) = {}
Merge(2) = { 2 }
Merge(3) = { 2 }
Merge(4) = { 2,5,6 }
Merge(5) = { 2,5,6 }
Merge(6) = { 2,5,6 }
Merge(7) = { 2 }
Merge(8) = { 2,5,6,8 }
Merge(9) = { 2,5,6,8 }
Merge(10) = { 2,5,6,8 }
Merge(11) = {}



How are Merge and T_q related ?

Observation: The Merge set of a node q denoted as Merge(q) and T_q are related by the following formula:

$$T_q - \{q\} \subseteq Merge(q)$$

$$T_9 = \{2,5,8,9\},\ Merge(9) = \{2,5,6,8\}$$

$$T_9 - \{9\} \subseteq Merge(9)$$

IsLiveIn and IsLiveInMergeSet

```
bool IsLiveIn(var a, node q){
    T_(q,a) ← T_q ∩ sdom(def(a))
    // sdom(x) contains all nodes strictly dominated by x
    for t in T_(q,a) do
        if R_t ∩ uses(a) ≠ Φ then return true
    return false
}
```

```
bool IsLiveInMergeSet(var a, node q){
    M_(q,a) ← ( Merge(q)U {q} ) ∩ sdom(def(a))
    for t in M_(q,a) do
        if R_t ∩ uses(a) ≠ Φ then return true
    return false
}
```

IsLiveIn= IsLiveInMergeSet

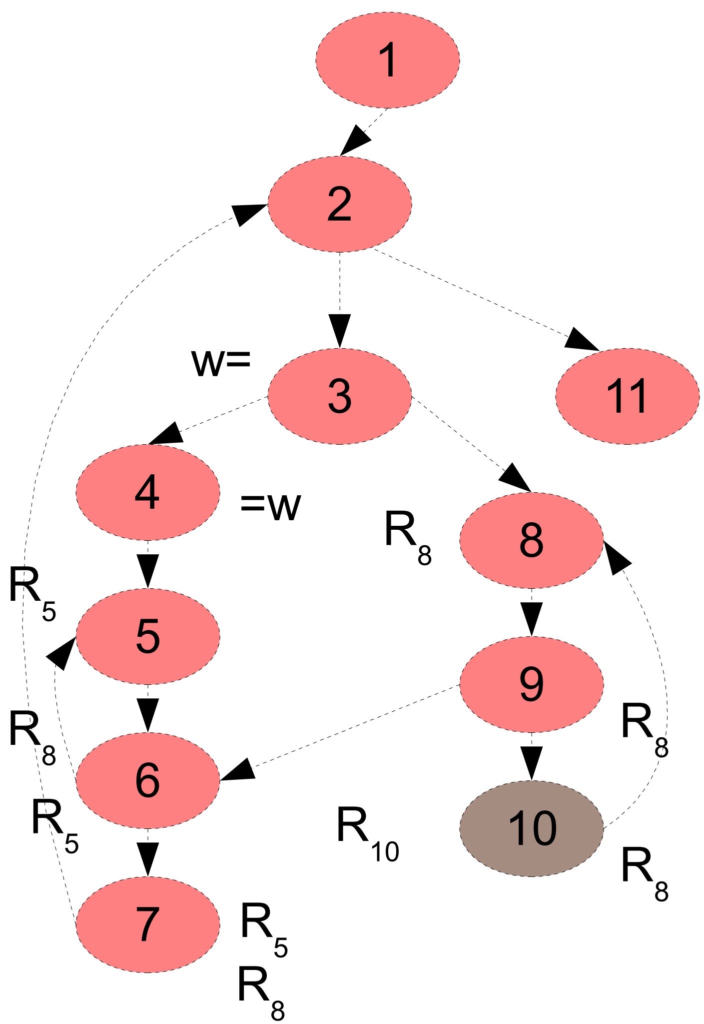
IsLiveInMergeSetUsingDJGraph

```
bool   IsLiveInMergeSetUsingDJGraph(q,a) {
     M(q,a) ← ( Merge(q)U {q} )
     for t in uses(a) do {
        while ( level(t) != level(def(a)) && t != def(a)) {
             if ( t ∩ M(q,a) )
                 return true
             t = dom-parent(t)
             //Climb up from node t in the DJ Graph
       } // end while
     } // end for
     return false
}
```

IsLiveIn= IsLiveInMergeSetUsingDJGraph

IsLiveIn vs IsLiveInMergeSetUsingDJGraph

- IsLiveIn(10,w) → False



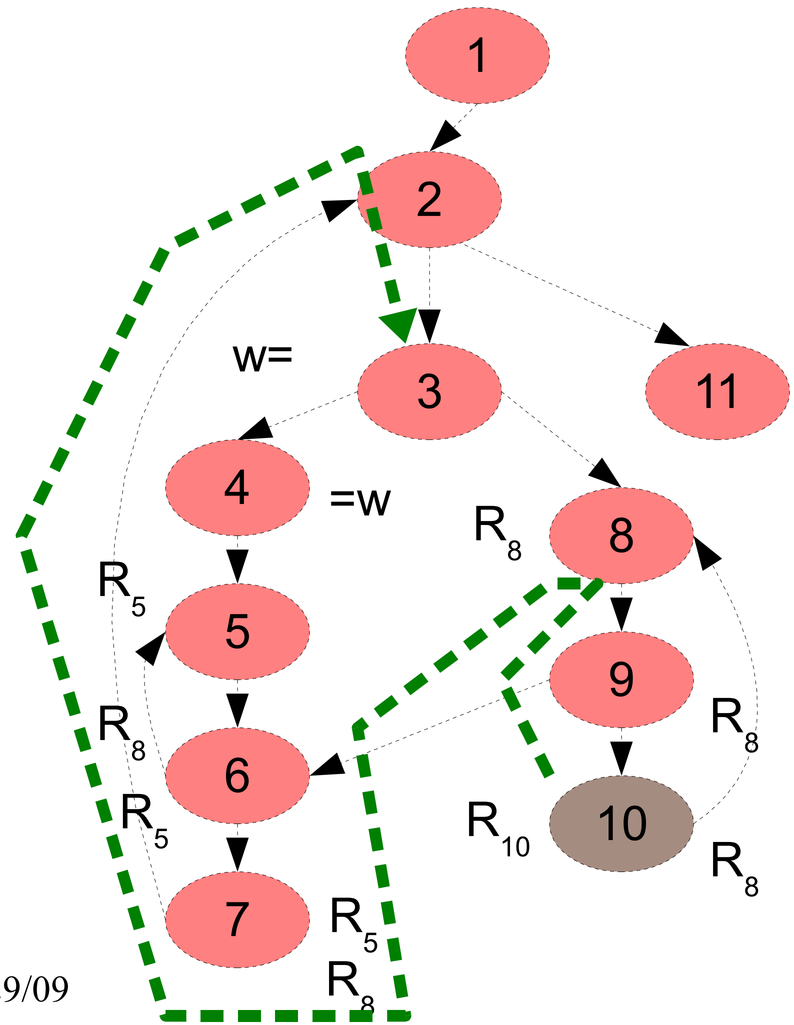
$$T_{10} = \{2,5,8,10\}$$
$$T_{10,w} = T_{10} \cap sdom(def(w))$$
$$= T_{10} \cap sdom(3)$$
$$= \{2,5,8,10\} \cap \{3 \ldots 10\}$$
$$= \{5,8,10\}$$

uses(w) = {4}

$R_5, R_8, R_{10} \cap \{4\} = \{\}$

IsLiveIn vs IsLiveInMergeSetUsingDJGraph

- IsLiveIn(10,w) → False



$T_{10} = \{2,5,8,10\}$

$T_{10,w} = T_{10} \cap sdom(def(w))$

$= T_{10} \cap sdom(3)$

$= \{2,5,8,10\} \cap \{3 \ldots 10\}$

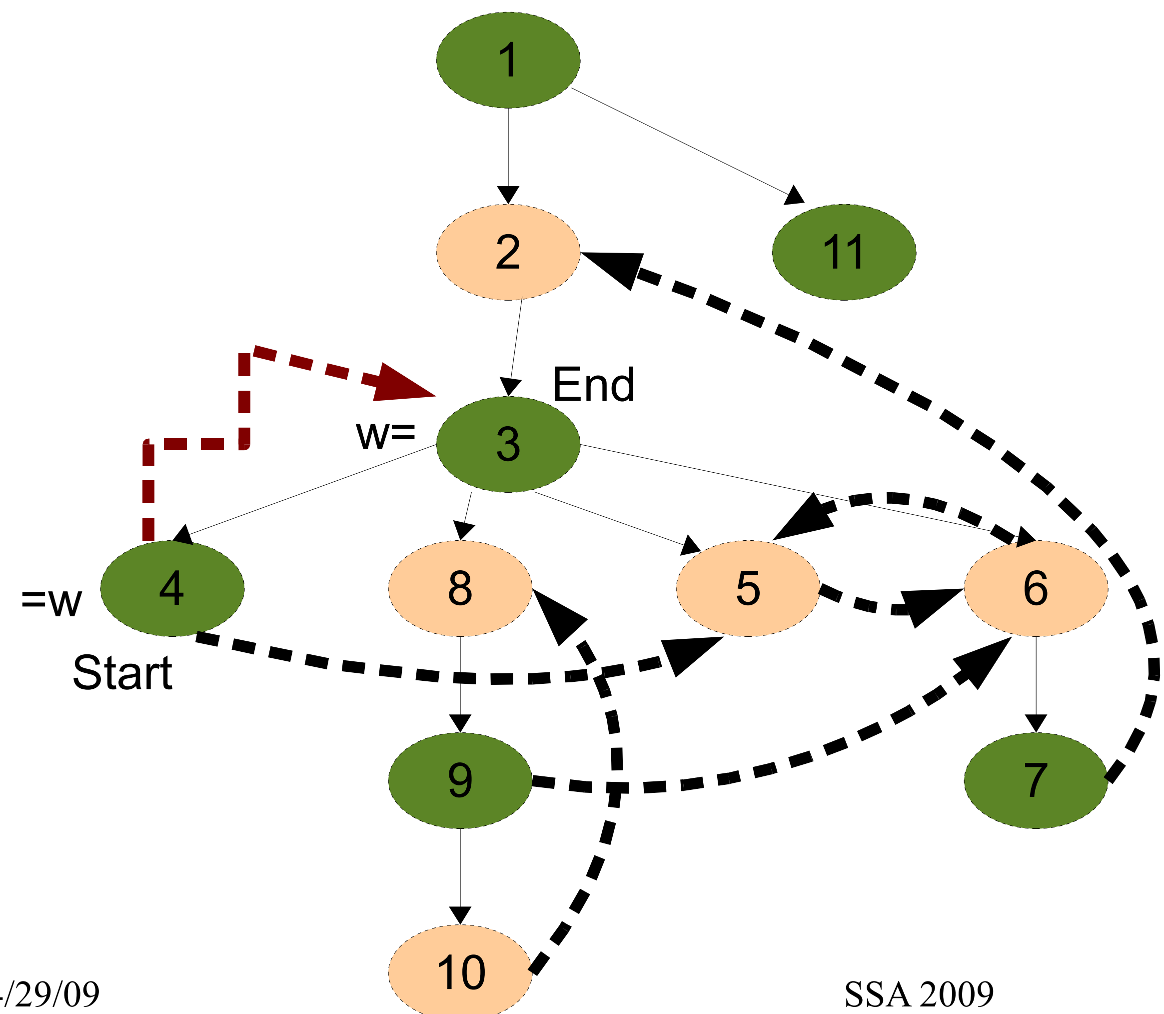
$= \{5,8,10\}$

uses(w) = {4}

$R_5, R_8, R_{10} \cap \{4\} = \{\}$

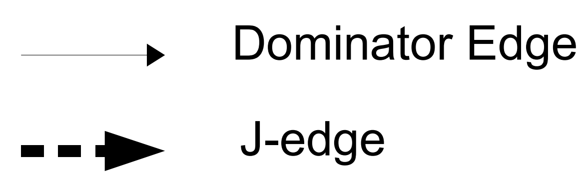
IsLiveIn vs IsLiveInMergeSetUsingDJGraph

- IsLiveInMergeSetUsingDJGraph(10,w) → False



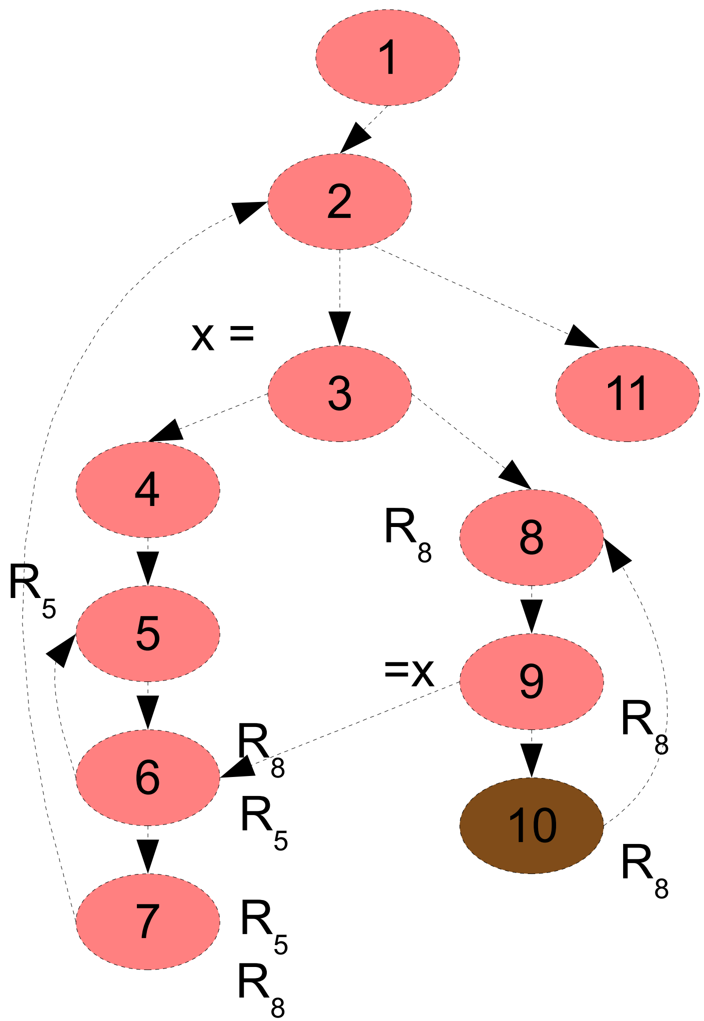
$$M_{10,w} = M_{10} \text{ U } \{10\}$$
$$= \{2,5,6,8\} \text{ U } \{10\}$$
$$= \{2,5,6,8,10\}$$

uses(w) = {4}
def(w) = {3}



IsLiveIn vs IsLiveInMergeSetUsingDJGraph

- IsLiveIn(10,x) → True



$T_{10} = \{2,5,8,10\}$

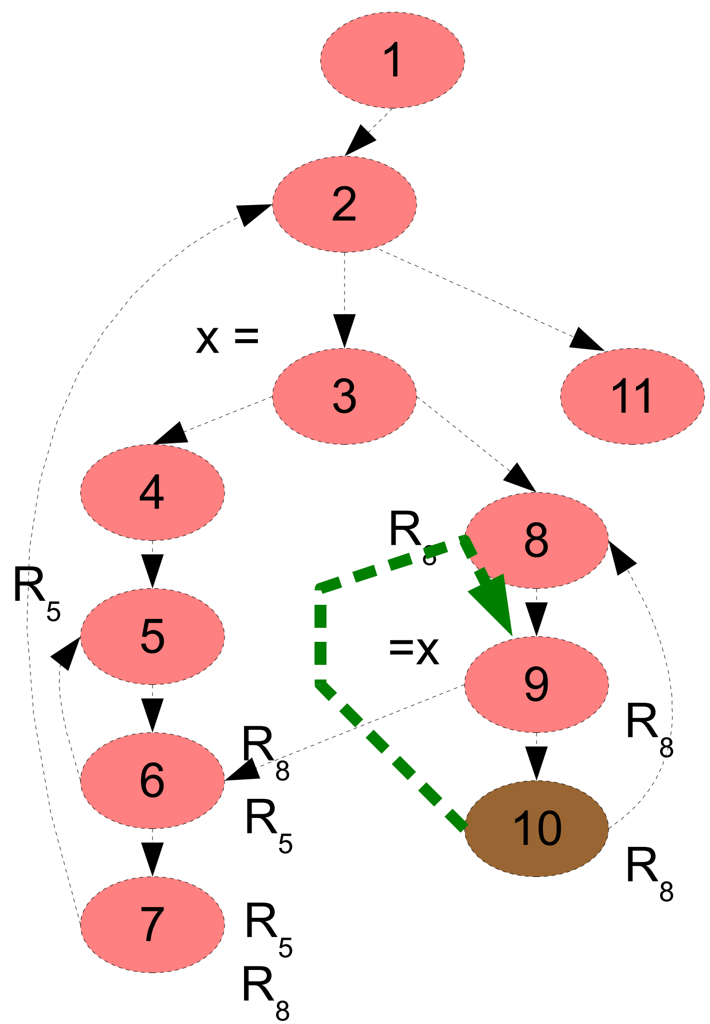
$T_{10,y} = T_{10} \cap \mathrm{sdom}(\mathrm{def}(x))$
$= T_{10} \cap \mathrm{sdom}(3)$
$= \{2,5,8,10\} \cap \{3 \ldots 10\}$
$= \{5,8,10\}$

uses(x) = {9}

$R_{10}, R_5, R_8 \cap \{9\} \neq \{\}$

IsLiveIn vs IsLiveInMergeSetUsingDJGraph

- IsLiveIn(10,x) → True



$T_{10} = \{2,5,8\}$

$T_{10,x} = T_{10} \cap sdom(def(x))$

$= T_{10} \cap sdom(3)$

$= \{2,5,8,10\} \cap \{3 \ldots 10\}$

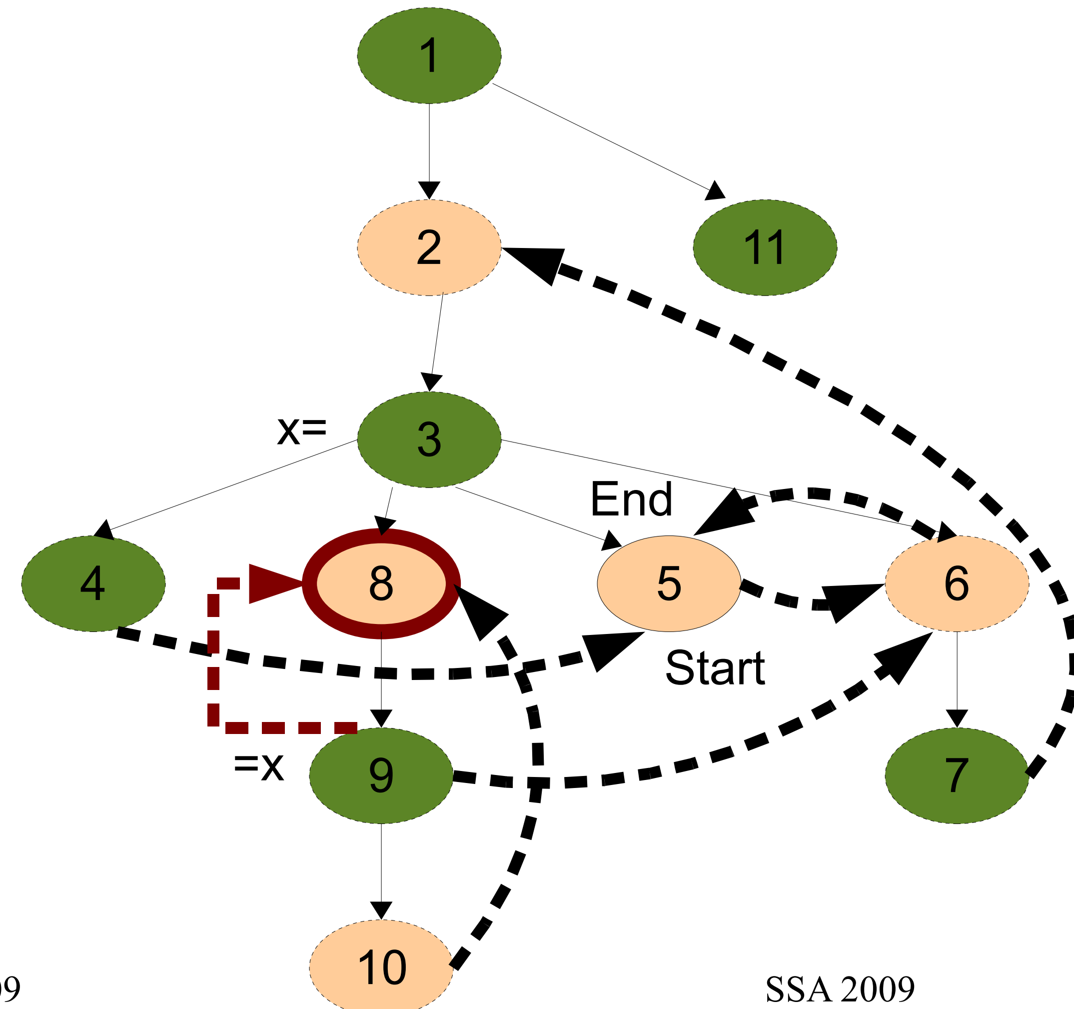
$= \{5,8,10\}$

uses(x) = {9}

$R_{10}, R_5, R_8 \cap \{9\} \neq \{\}$

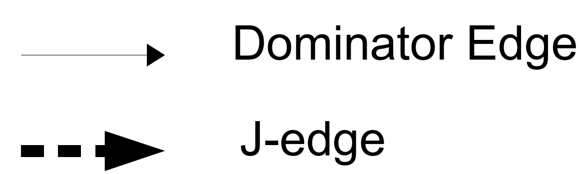
IsLiveIn vs IsLiveInMergeSetUsingDJGraph

- IsLiveInMergeSetUsingDJGraph(10,x) → True



$M_{10,w}$ = M_{10} U {10}
= {2,5,6,8} U {10}
= {2,5,6,8,10}

uses(x) = {9}
def(y) = {3}



Conclusion

- New algorithm for handling Liveness Analysis using DJ Graphs
- Simplified handling via the usage of Merge Sets - removes the need of computing T_q and R_q sets
- Merge Sets can be computed easily for both reducible and irreducible graphs efficiently